the globus alliance
www.globus.org

The Globus eXtensible Input/ Output System (XIO): A protocol independent IO system for the Grid

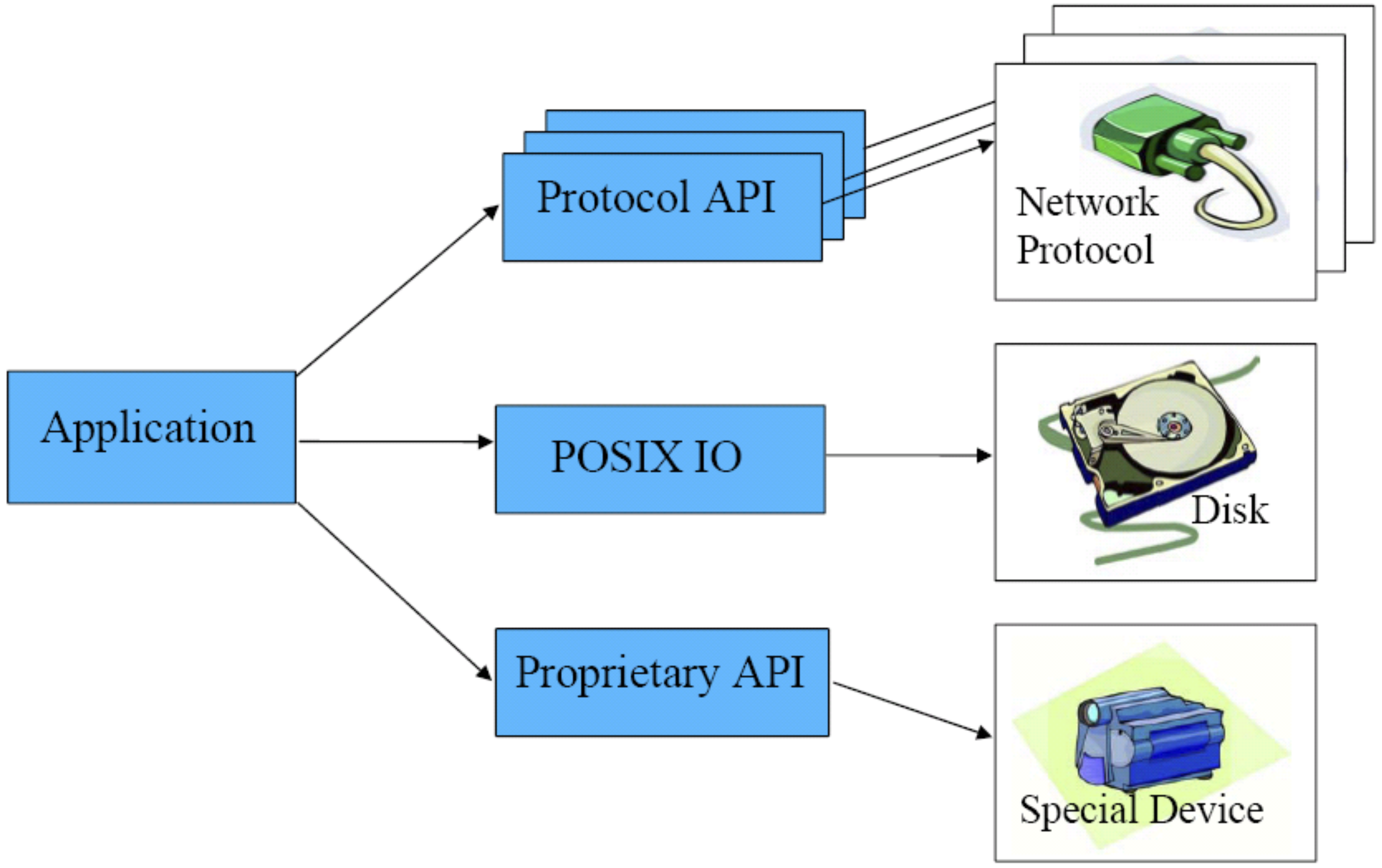
Bill Allcock, John Bresnahan, Raj Kettimuthu and Joe Link

Argonne National Laboratory

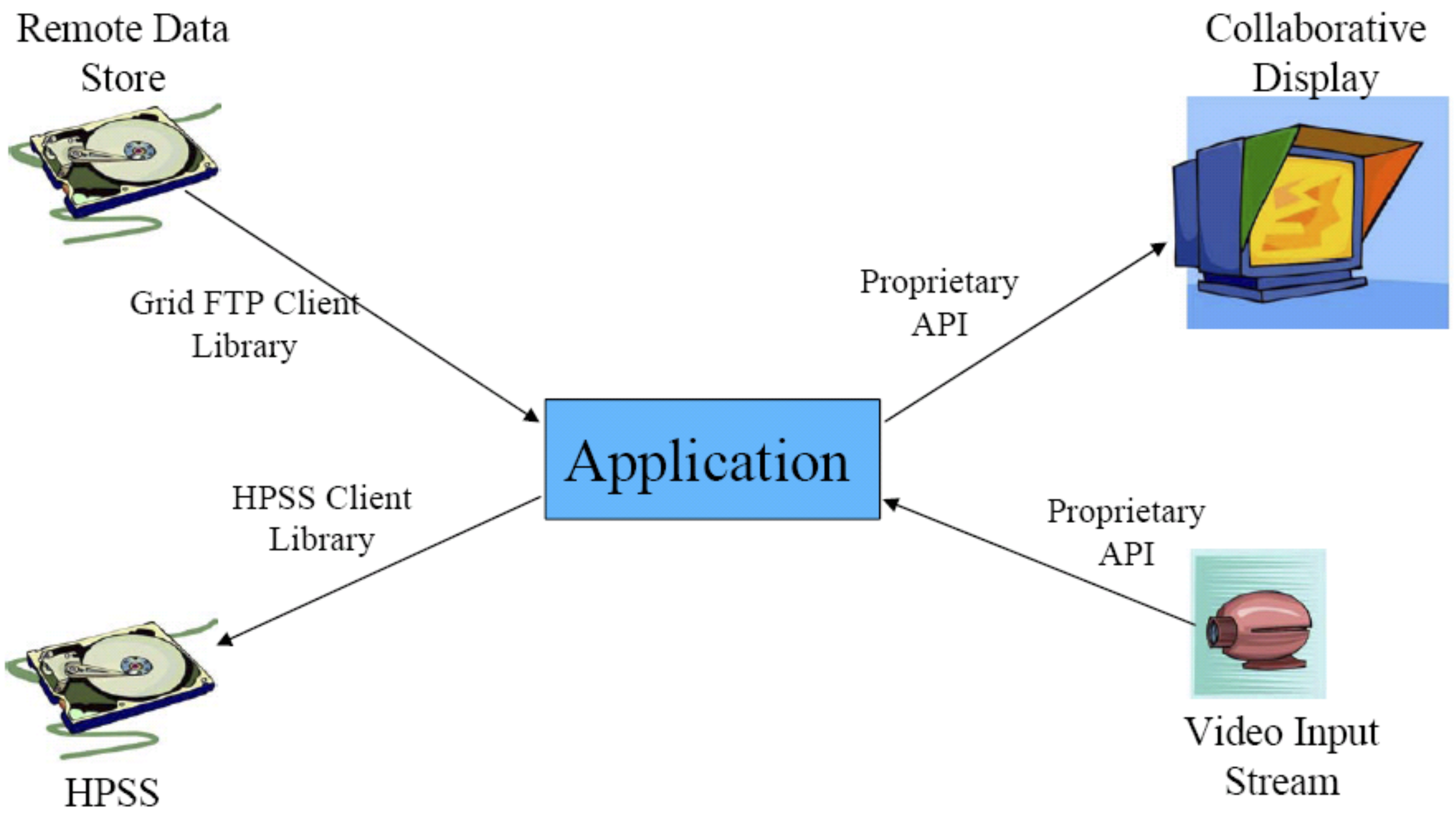
Grid IO

- Geographically disperse resources
  - Supercomputers/clusters
  - Large data store
  - Specialized scientific devices
    - Telescopes, environmental sensors
  - Collaborative sessions
    - Access Grid
- Varying network and file transfer protocols
  - TCP, UDP, XCP, GridFTP etc

Typical Approach



Example Application



the globus alliance
www.globus.org

Problems

- Development time
  - Application must learn to use many different APIs
  - Each API has its own semantics
    - Asynchronous/Synchronous
    - Threaded/non-threaded
- Scalability and compatibility
  - Application must be modified to work with new protocols/devices
  - Application must keep up with orthogonal research issues

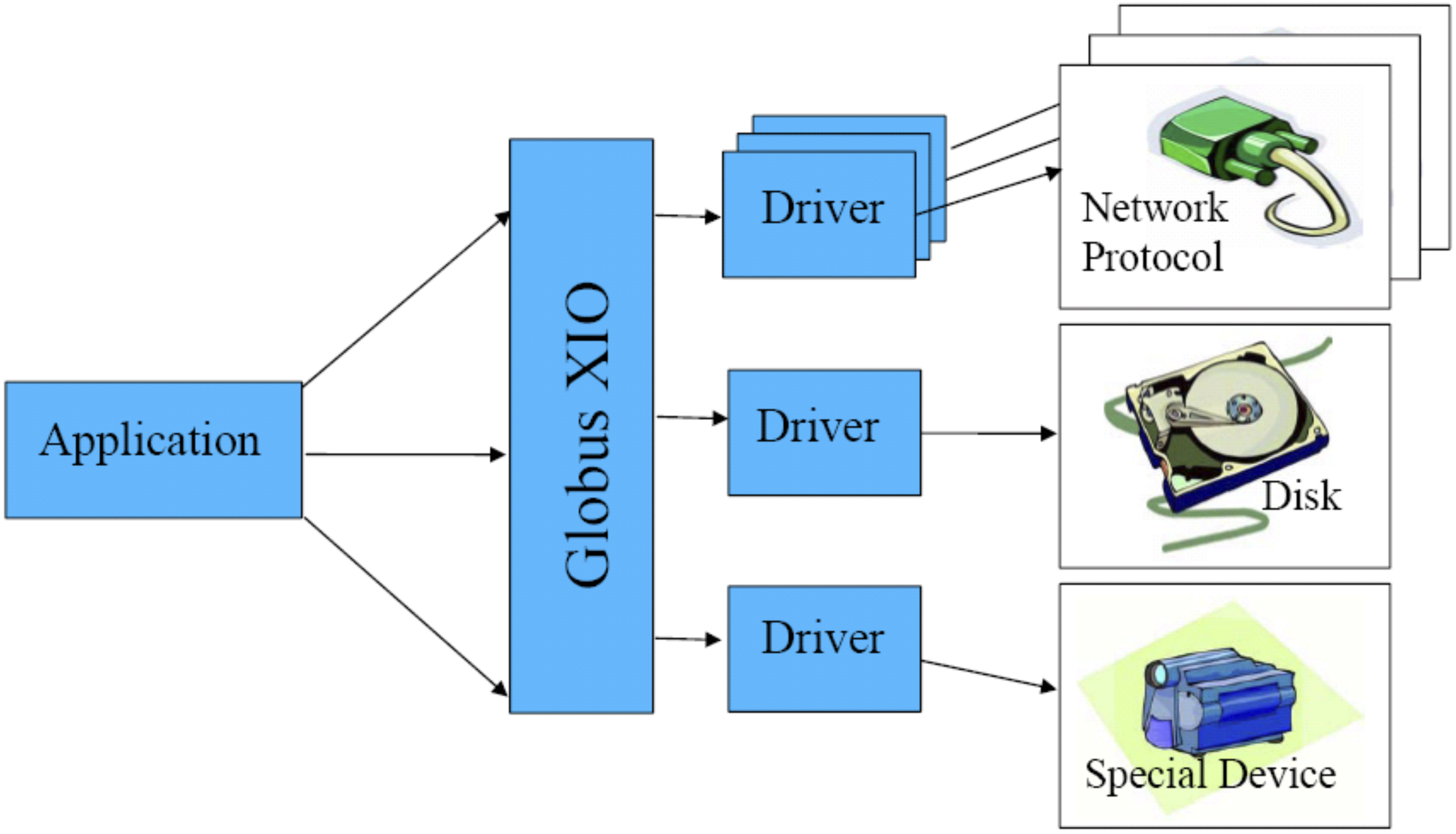
Observations

- Data stream IO abstraction
  - Many Grid IO needs can be treated as a stream of bytes
  - Open/close/read/write functionality satisfies most requirements
- Protocol details
  - Rarely does the application need to deal with protocol details
  - Most needs can be satisfied at initialization time

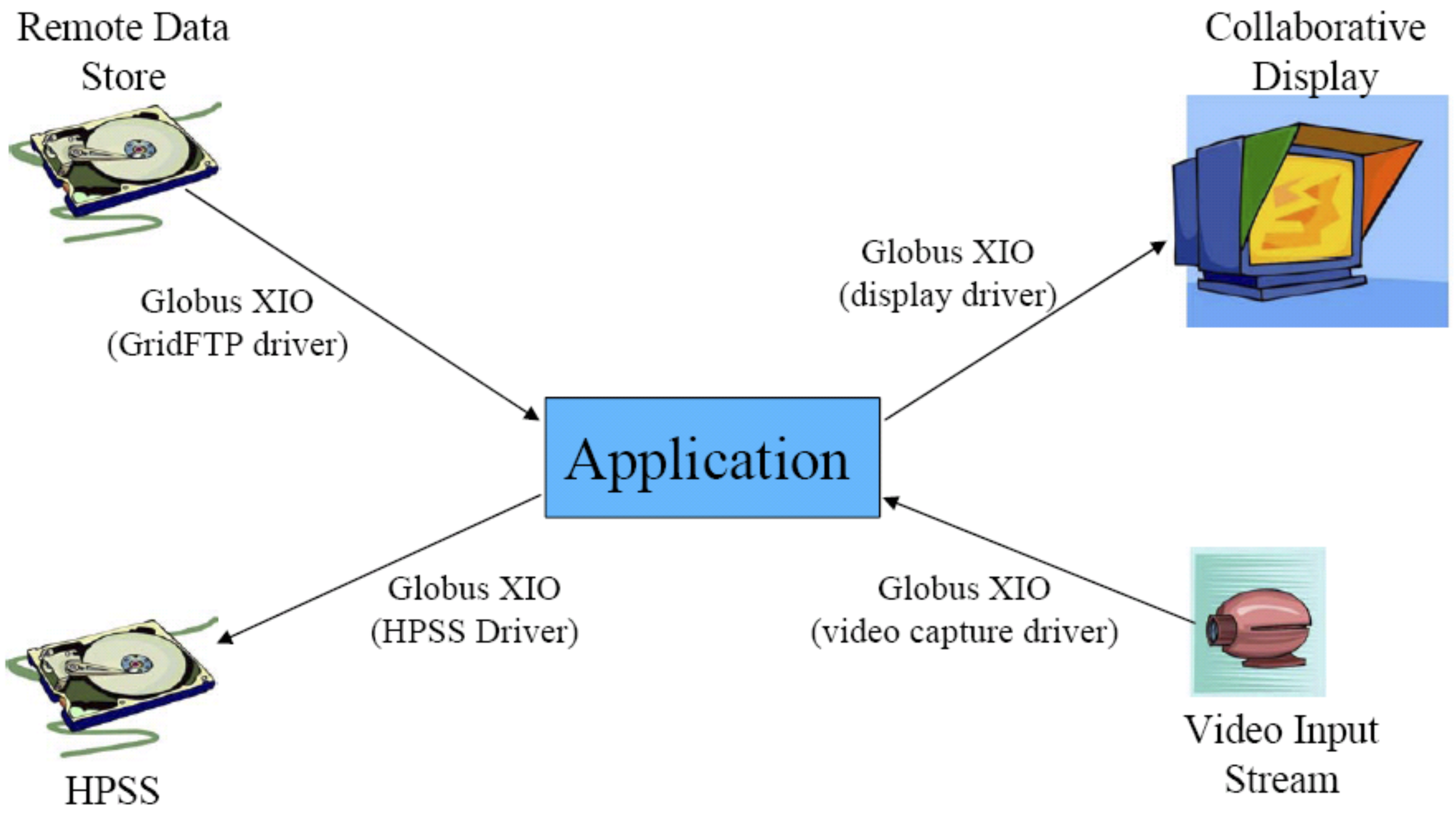
Solution

- Globus XIO user API
  - Single API/single set of semantics
  - Simple open/close/read/write API
- Driver abstraction
  - Hides protocol details
  - Allows for extensibility
  - Drivers can be selected at runtime

Globus XIO Approach



Example Application with XIO

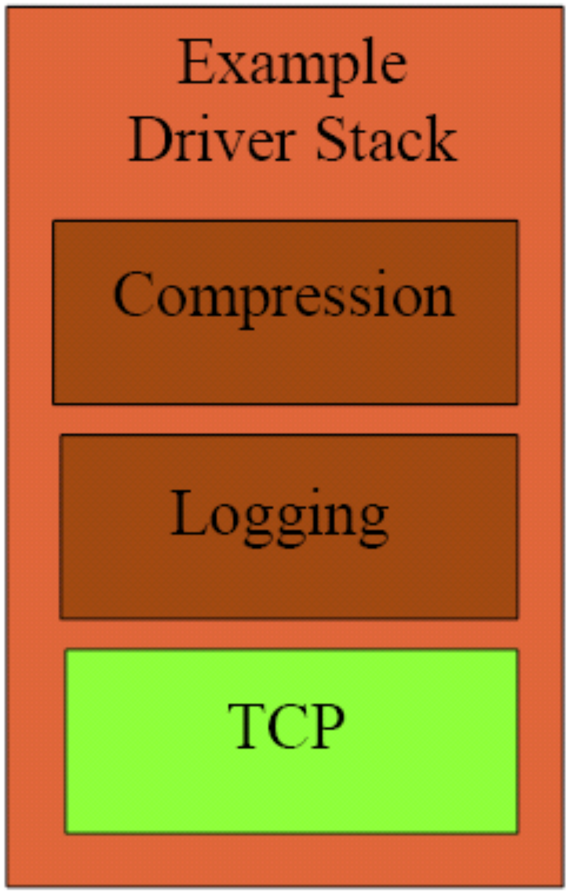


Drivers

- Make single API do many types of IO
- Specific drivers for specific protocols/devices
- Transport
  - Moves data out of process space
  - TCP, UDP, File etc
- Transform
  - Manipulate or examine data
  - Do not move data outside of process space
  - Compression, security etc

Stack

- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack
- Control flows from user to the top of the stack, to the transport driver



Driver development

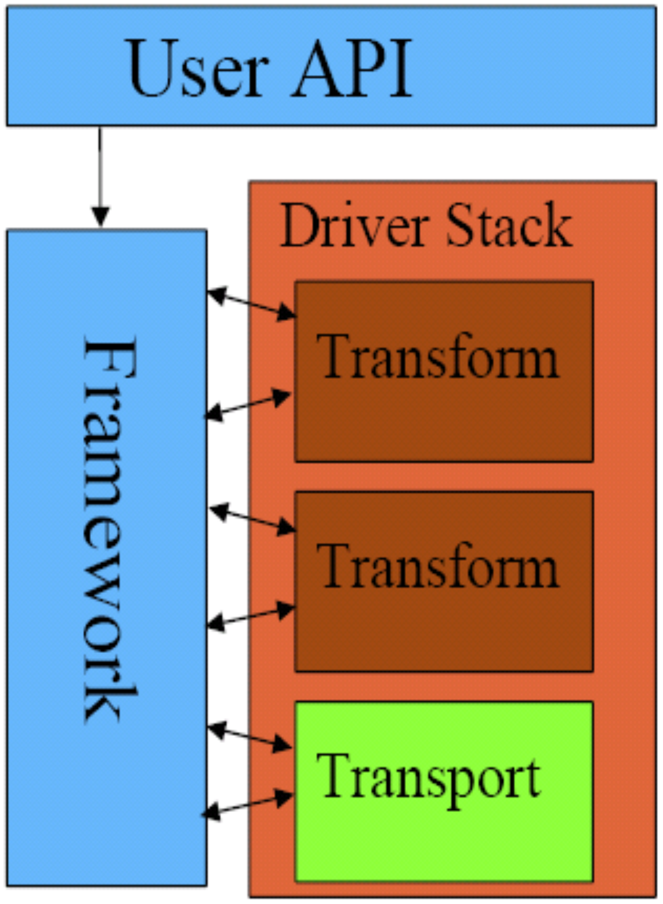
- A set of function signatures
  - Implemented by the driver
  - Registered with Globus XIO
- Semantics
  - XIO makes calls to these functions expecting specific behavior
  - Read() interface function should produce data and write() should consume data

Attributes and controls

- Way to expose functionalities specific to a protocol
- User API to control the attributes and special features of a driver
  - An opportunity to tweak parameters specific to a driver
- Attributes and controls support is optional and a driver may choose not to have it

Globus XIO Framework

- Moves data from user to driver stack
- Manages the interactions between drivers
- Assist in the creation of drivers
    - Asynchronous support
    - Close and EOF barriers
    - Error checking
    - Internal API for passing operations down the stack



Existing drivers

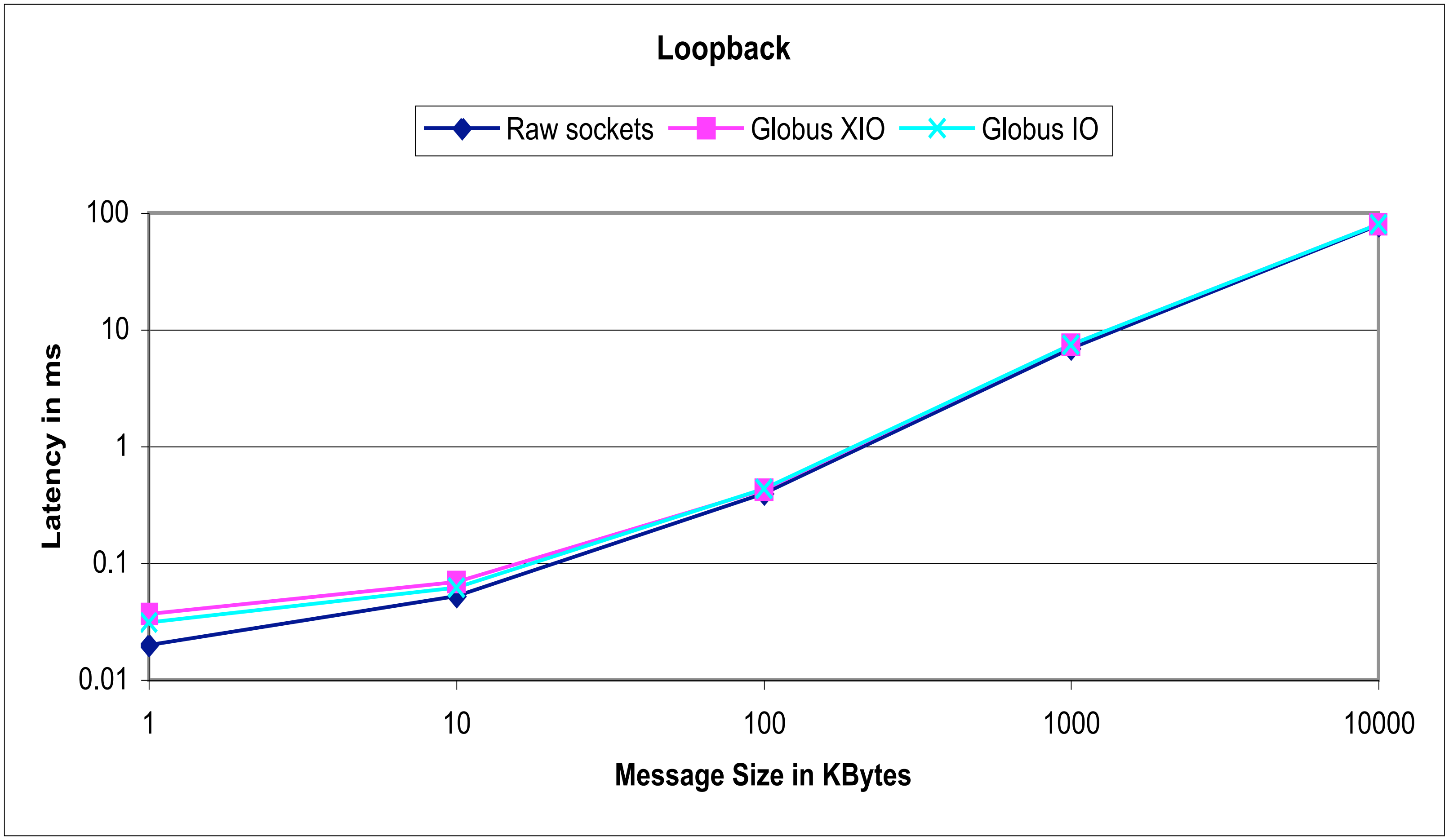
- Transport
  - TCP, UDP, File, GridFTP
  - UDT - UDP based reliable data transport protocol
  - Multi-stream - uses multiple transport protocol streams for data transfer
- Transform
  - GSI, HTTP, Ordering

Performance analysis

- Compare with raw sockets and Globus IO
- Latency tests
  - Ping-Pong tests, 1000 times, average of 10 runs
  - Loopback, LAN and WAN
- Bandwidth tests
  - Send 1000 back-to-back messages and wait for reply, average of 10 runs
  - LAN and WAN

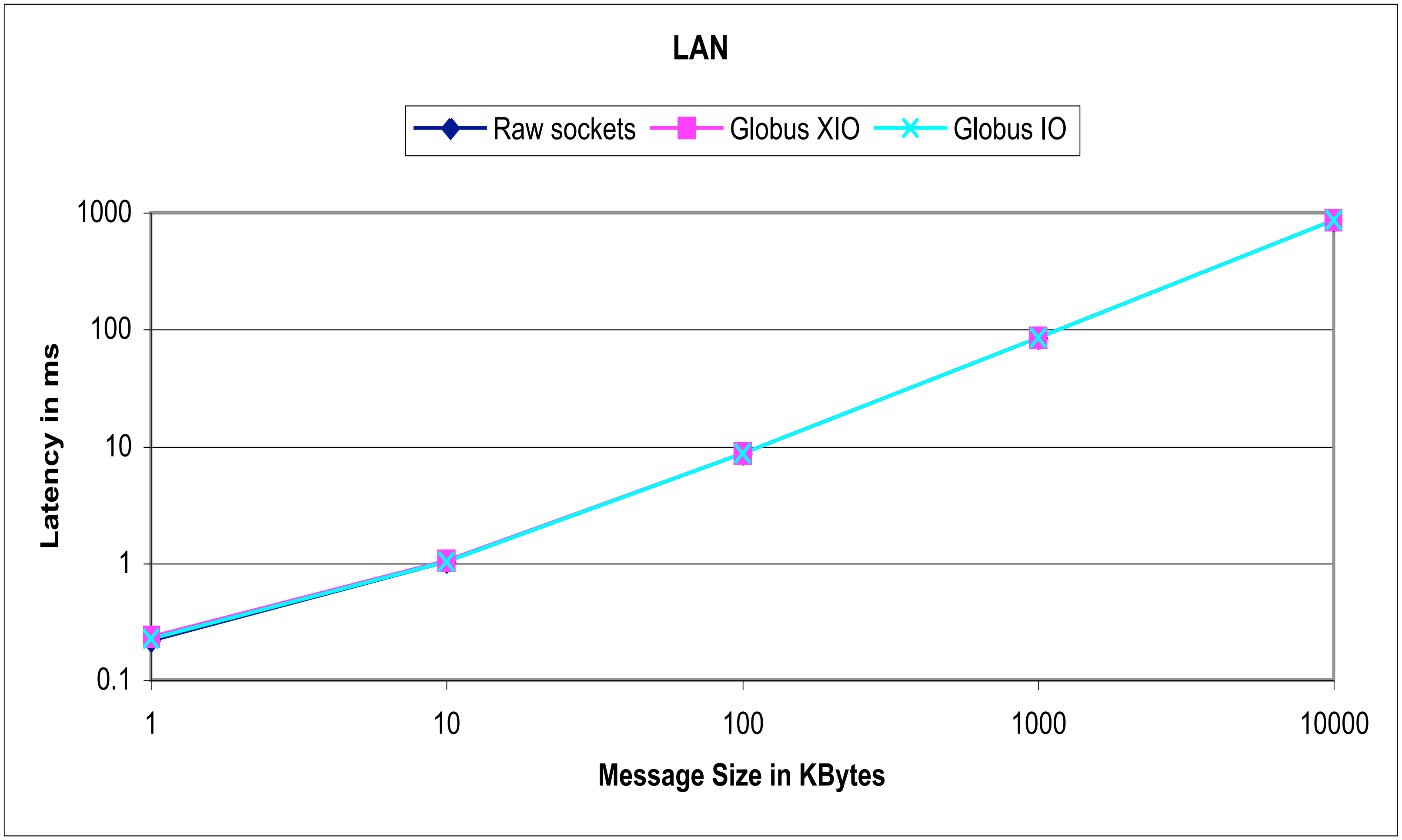
the globus alliance
www.globus.org

Latency over Loopback Interface



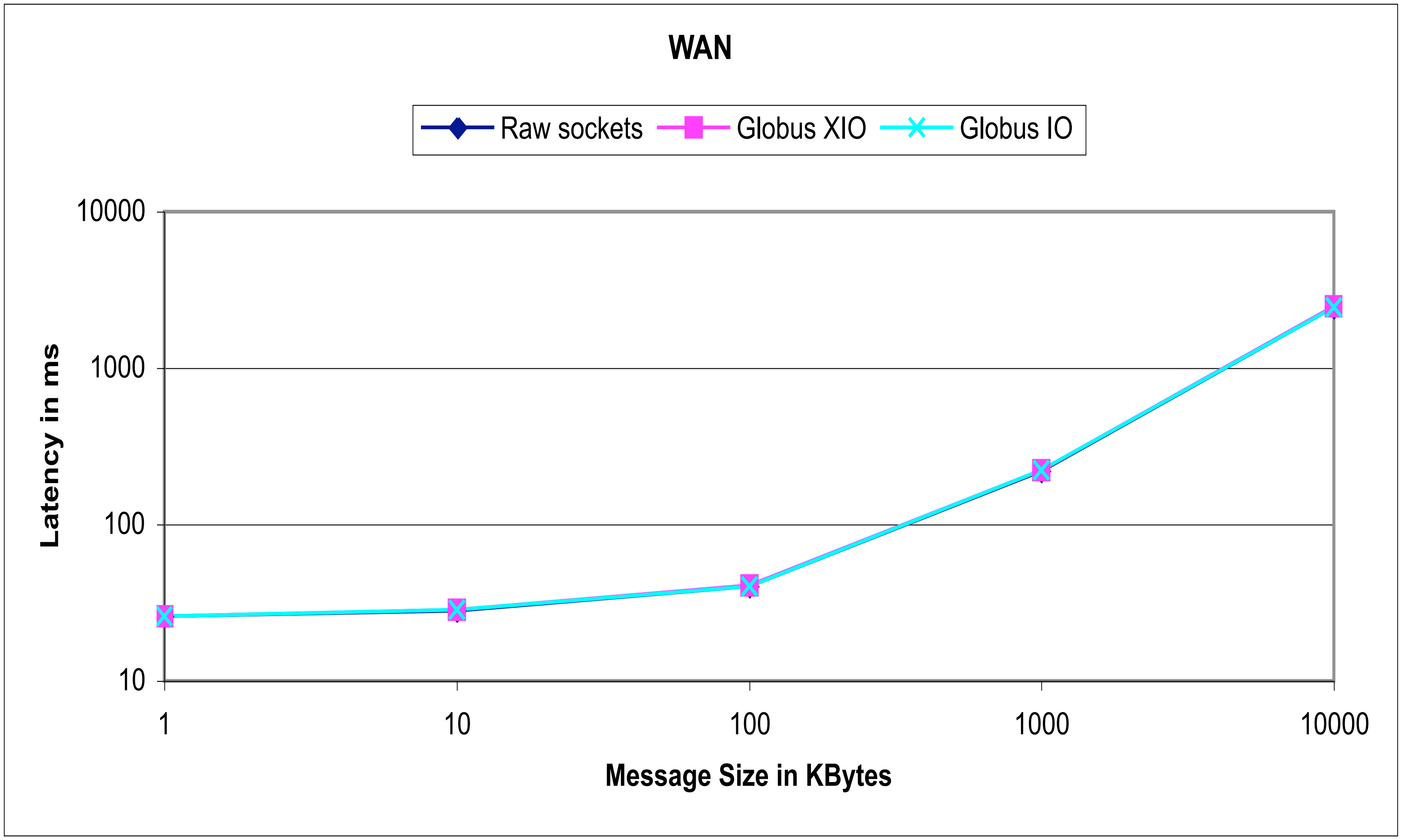
the globus alliance
www.globus.org

Latency comparison on LAN



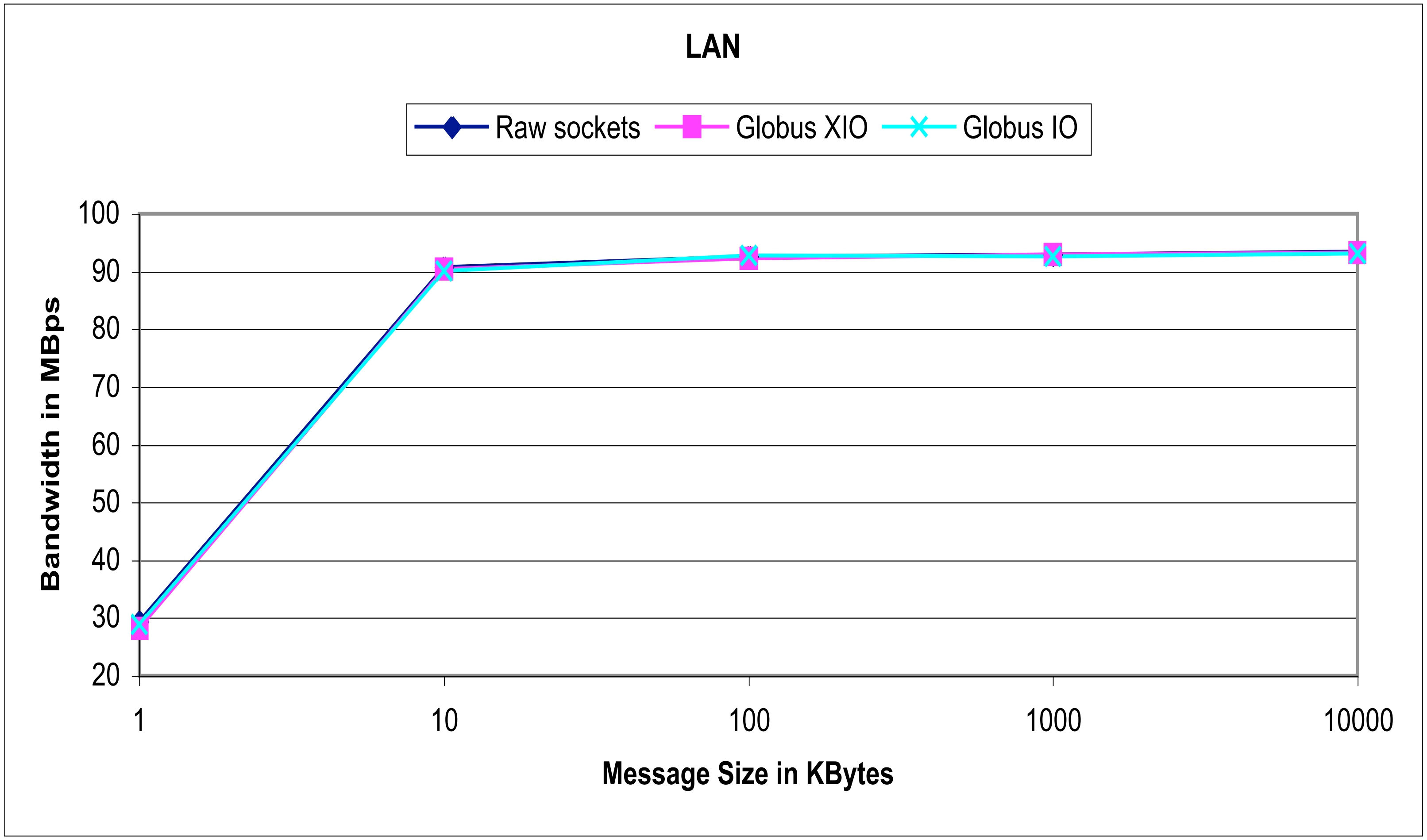
the globus alliance
www.globus.org

Latency comparison on a WAN



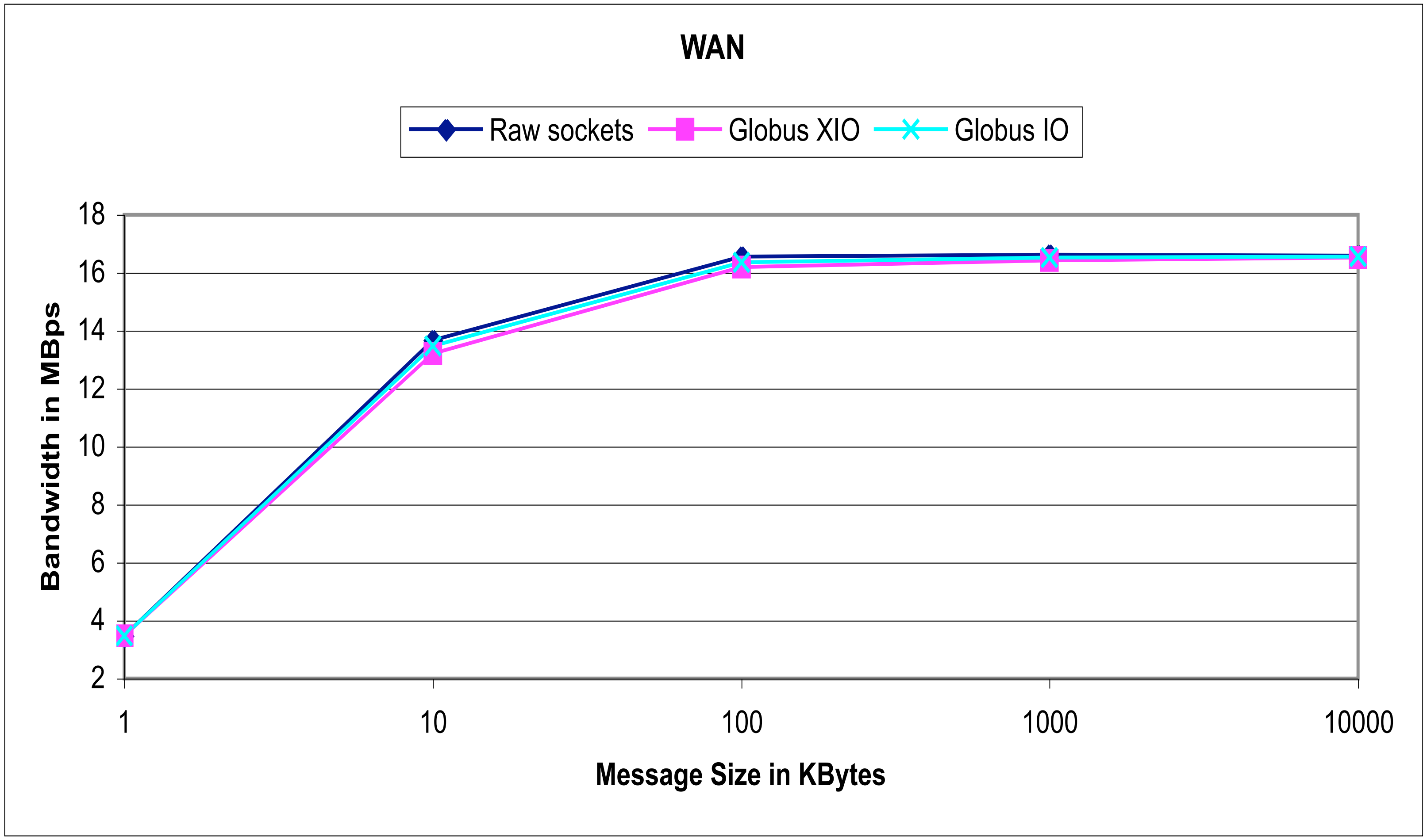
the globus alliance
www.globus.org

Bandwidth on a LAN



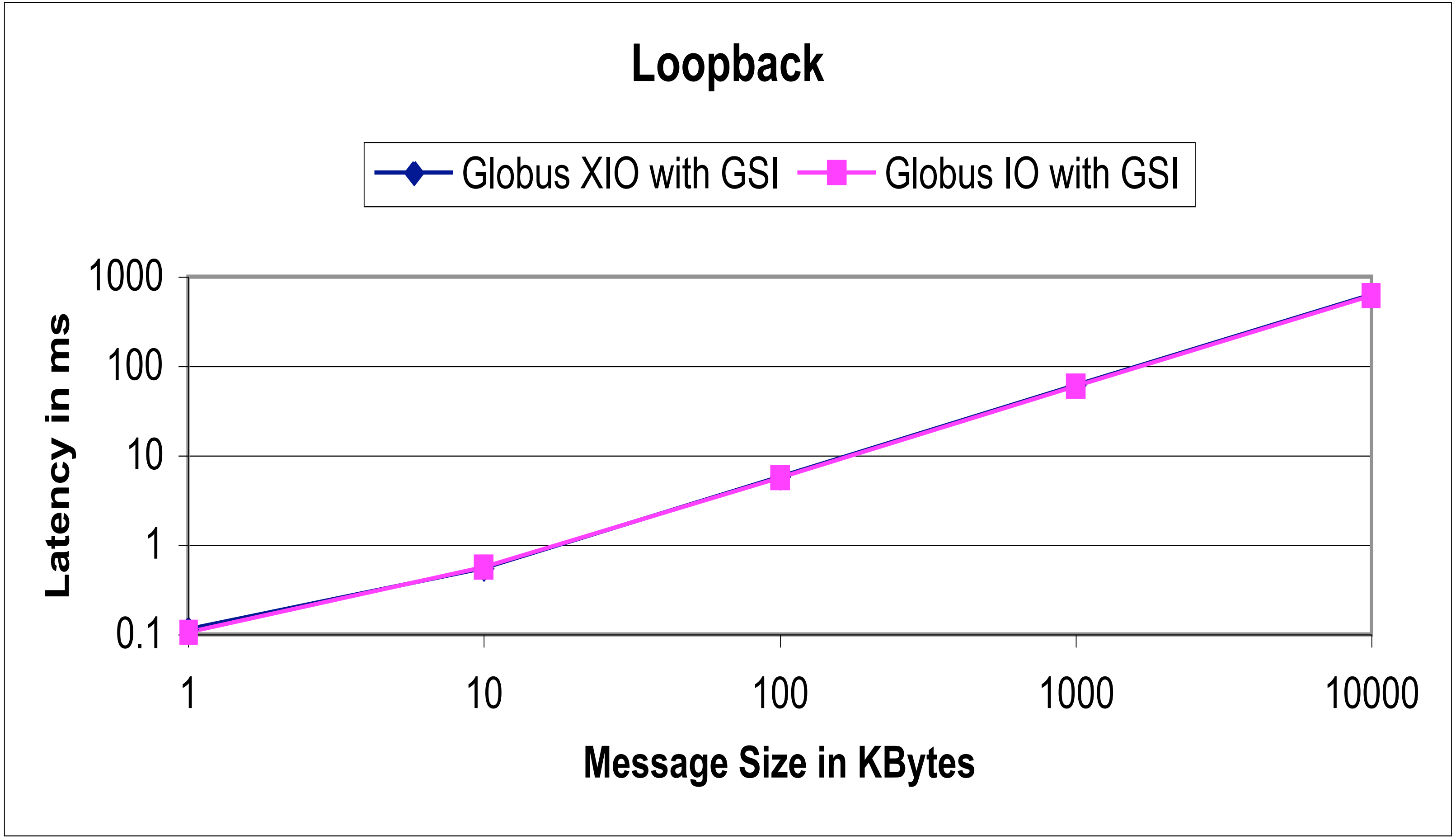
the globus alliance
www.globus.org

Bandwidth on a WAN



the globus alliance
www.globus.org

Multiple drivers on a stack



the globus alliance
www.globus.org

Summary

- Provides a simple and uniform IO API
  - Appropriate for most byte stream oriented applications
- Ability to quickly adapt to
  - Different transport protocols
  - Different data access mechanisms
- Provides a convenient framework for research and development